EAST Search History

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L2	0	"200398478"	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/05/28 21:08
L3	2	"2002145571"	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/05/28 21:09
L4	3	(optimized adj database adj appliance).ti.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/05/28 21:10
L5	3	(optimized adj database adj appliance).ti. and (error fail$4)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/05/28 21:14
L6	2	"20050097078"	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/05/28 21:16
L7	52	(query with check$1point)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/05/28 21:16
L8	9	(query with check$1point) and ((stop$4 exit$4 quit$4 terminat$4 halt$4) with execution)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/05/28 21:56

EAST Search History

L9	0	(query with check$1point) same ((stop$4 exit$4 quit$4 terminat$4 halt$4) with failure)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/05/28 21:25
L10	10	(query with (check$1point check)) same ((stop$4 exit$4 quit$4 terminat$4 halt$4) with execution)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/05/28 21:25
L11	9	L10 not L8	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/05/28 21:25
L12	0	"20050097078" and (iterative (another with error))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/05/28 21:56
L13	26	707/100.ccls. and ((query model plan QEP) with error with (run$1time execut$4))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/05/28 22:04
S1	1339	quer$4 and ((access$3 execut$4) with plan) and ((halt$3 quit$3 end$3 stop$4 terminat$4) with (query plan))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/01/02 12:22
S2	3	quer$4 and ((access$3 execut$4) with plan) and ((halt$3 quit$3 end$3 stop$4 terminat$4) with (query plan) with execut$4 with error)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/01/02 12:37

EAST Search History

S3	3	quer$4 and ((access$3 execut$4 query) with plan) and ((halt$3 quit$3 end$3 stop$4 terminat$4) with (query plan) with execut$4 with error)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/01/03 11:31
S4	3	quer$4 and ((access$3 execut$4 query process$3) with plan) and ((halt$3 quit$3 end$3 stop$4 terminat$4) with (query plan) with execut$4 with error)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/01/02 12:37
S5	5	quer$4 and ((access$3 execut$4 query) with (plan or model)) and ((halt$3 quit$3 end$3 stop$4 terminat$4) with (query plan) with execut$4 with error)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/01/03 11:46
S6	44	quer$4 and ((access$3 execut$4 query process) with (plan or model)) and ((halt$3 quit$3 end$3 stop$4 terminat$4) with (query plan) with error)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/01/03 11:49
S7	6	quer$4 and ((access$3 execut$4 query process) with (plan or model)) and (((halt$3 quit$3 end$3 stop$4 terminat$4) with (query plan) with error) same execut$4)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/01/03 11:49
S8	17	quer$4 and ((access$3 execut$4 query process) with (plan or model)) and (((halt$3 quit$3 end$3 stop$4 terminat$4) with (query plan) with error) same (execut$4 run$4 operat$3))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/01/03 11:59
S9	0	quer$4 and ((access$3 execut$4 query process) with (plan or model)) and (((halt$3 quit$3 end$3 stop$4 terminat$4) with (query plan) with error) near (execut$4 run$4 operat$3))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/01/03 11:51

EAST Search History

S10	1	(((self$heal$3) or (self adj heal$3)) with engine) and ((quer$4 plan) with error)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/01/03 11:56
S11	2	(((self$heal$3) or (self adj heal$3)) with engine) and ((quer$4 plan))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/01/03 11:56
S12	2	quer$4 and ((access$3 execut$4 query process) with (plan or model)) and (((halt$3 quit$3 end$3 stop$4 terminat$4) with (query plan) with error) same (execut$4 run$4 operat$3)) and ((re$build$3 or re$creat$3 or auto$tun$3 or re$generat$4) with (query plan model))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/01/03 12:10
S13	41	microsoft.as. and ((self$heal$3) or (self adj heal$3))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/01/03 12:12
S14	9	microsoft.as. and (((self$heal$3) or (self adj heal$3)) with (engine software))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/01/03 12:11
S15	7	microsoft.as. and (((self$heal$3) or (self adj heal$3)) with (engine software)) and quer$4	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/01/03 12:11
S16	5	microsoft.as. and ((self$heal$3) or (self adj heal$3)) and quer$4 and ((access$3 execut$4 query) with (plan or model))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/01/03 12:13

EAST Search History

S17	45	(error with (hand$3 report$3)) and ((execution run$1time) with query with (error failure))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/05/28 18:33
S18	33	(error with (hand$3 report$3)) and ((execution run$1time) with query with (error failure)) and sql	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/05/28 18:37
S19	3	(error with (hand$3 report$3)) and (((execution run$1time) with query with (error failure)) same ((halt$3 stop$4 terminat$4 (early adj out) early$1out) with (query plan execut$4 run$1time))) and sql	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/05/28 18:45
S20	3	(error with (hand$3 report$3)) and (((execution run$1time) with (query plan QEP) with (error failure)) same ((halt$3 stop$4 terminat$4 (early adj out) early$1out) with (query plan execut$4 run$1time))) and sql	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/05/28 18:46
S21	14	(error with (hand$3 report$3)) and (((execution run$1time) with (query plan QEP) with (error failure)) and ((halt$3 stop$4 terminat$4 (early adj out) early$1out) with (query plan execut$4 run$1time))) and sql	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/05/28 19:03
S22	4	"6757671"	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/05/28 19:03
S23	0	"6757671" and (error and "early out")	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/05/28 19:04

EAST Search History

S24	0	"6757671" and (error and early$1out)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/05/28 19:04
S25	0	"6757671" and early$1out	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/05/28 19:06
S26	0	((query model plan QEP) with error with early$1out)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/05/28 19:07
S27	27567	((query model plan QEP) with error)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/05/28 19:07
S28	921	((query model plan QEP) with error with (run$1time execut$4))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/05/28 22:04
S29	216	((query model plan QEP) with error with (run$1time execut$4)) and (error with (halt$3 terminat$4 stop$4 exit$3 quit$3))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/05/28 19:15
S30	41	"707"/.ccls. and ((query model plan QEP) with error with (run$1time execut$4)) and (error with (halt$3 terminat$4 stop$4 exit$3 quit$3))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/05/28 21:08

[File 348] **EUROPEAN PATENTS** 1978-2007/ 200718

**File 348: For important information about IPCR/8 and forthcoming changes to the IC= index, see HELP NEWSIPCR.*

[File 349] **PCT FULLTEXT** 1979-2007/UB=20070510UT=20070504

**File 349: For important information about IPCR/8 and forthcoming changes to the IC= index, see HELP NEWSIPCR.*

; d s

Set	Items	Postings	Description
S1	591	4338	S (QUERY OR QUERIES)(3N)(OPTIMIZ??? OR OPTIMIS??? OR OPTIMIZATION OR OPTIMISATION)
S2	1842	13845	S (QUERY OR EXECUTION)(3N)PLAN? ? OR QAP OR QEP
S3	562861	5143248	S ERROR? ? OR FAIL???? OR FAULT? ? OR FUNCTION()CHECK???
S4	27972	129152	S (S2 OR PLAN? ? OR EXECUT???)(3N)(HALT??? OR STOP???? OR CEAS??? OR ABORT??? OR END???)
S5	18720	104362	S (PLAN? ? OR QAP OR QEP)(3N)(REBUILT OR REBUILD??? OR RE()(BUILT OR BUILD???) OR CORRECT??? OR MODIFY??? OR MODIFIE? ? OR MODIFICATION OR CHANG??? OR AMEND??? OR ALTER??? OR ALTERATION? ? OR EDIT??? OR REOPTIMIS? OR REOPTIMIZ?)
S6	9	29	S (MIDQUERY OR MID()QUERY OR SELFHEAL??? OR SELF()HEAL???)(3N)(REOPTIMIS? OR REOPTIMIZ? OR OPTIMIZ??? OR OPTIMIS??? OR OPTIMIZATION OR OPTIMISATION)
S7	3	27	S S1(50N)S2(50N)S4
S8	9	138	S S1(50N)S2(50N)S3
S9	8	108	S S1(50N)S2(50N)S5
S10	24	293	S S6:S9
S11	20	236	S S10 NOT AD=20040108:20070518/PR
S12	25	274	S S1:S2(100N)S3(100N)S4
S13	1	17	S S12(100N)S5
S14	24	257	S S12 NOT (S11 OR S13)
S15	18	192	S S14 NOT AD=20040108:20070518/PR

STIC Search

11/3K/6 (Item 4 from file: 349) **Links**
PCT FULLTEXT

01201071
OPTIMIZED SQL CODE GENERATION
GENERATION D'UN CODE SQL OPTIMISE

Patent Applicant/Patent Assignee:

- **NETEZZA CORPORATION**; 200 Crossing Boulevard, Framingham, MA 01702
  US; US(Residence); US(Nationality)
  (For all designated states except: US)
- **ZANE Barry M**; 4 Cobblestone Circle, Wayland, MA 01778
  US; US(Residence); US(Nationality)
  (Designated only for: US)
- **BALLARD James P**; 379 Fowler Road, Northbridge, MA 01534
  US; US(Residence); US(Nationality)
  (Designated only for: US)
- **HINSHAW Foster D**; 22 Campbell Park, Somerville, MA 02144
  US; US(Residence); US(Nationality)
  (Designated only for: US)
- **YERABOTHU Premanand**; 63 Latisquama Road, Southborough, MA 01772
  US; US(Residence); IN(Nationality)
  (Designated only for: US)
- **KIRKPATRICK Dana A**; 146 Whitney Street, Northborough, MA 01532
  US; US(Residence); US(Nationality)
  (Designated only for: US)

Patent Applicant/Inventor:

- **ZANE Barry M**
  4 Cobblestone Circle, Wayland, MA 01778; US; US(Residence); US(Nationality); (Designated only for: US)
- **BALLARD James P**
  379 Fowler Road, Northbridge, MA 01534; US; US(Residence); US(Nationality); (Designated only for: US)
- **HINSHAW Foster D**
  22 Campbell Park, Somerville, MA 02144; US; US(Residence); US(Nationality); (Designated only for: US)
- **YERABOTHU Premanand**
  63 Latisquama Road, Southborough, MA 01772; US; US(Residence); IN(Nationality); (Designated only for: US)
- **KIRKPATRICK Dana A**
  146 Whitney Street, Northborough, MA 01532; US; US(Residence); US(Nationality); (Designated only for: US)

Legal Representative:

- **WAKIMURA Mary Lou(et al)(agent)**
  Hamilton, Brook, Smith & Reynolds, P.C., 530 Virginia Road, P.O. Box 9133, Concord, MA 01742-9133; US;

	Country	Number	Kind	Date
Patent	WO	200508529	A2-A3	20050127
Application	WO	2004US21672		20040707

Priorities	US	2003485321		20030707
	US	2003485638		20030708

Designated States: (All protection types applied unless otherwise stated - for applications 2004+)
AE; AG; AL; AM; AT; AU; AZ; BA; BB; BG;
BR; BW; BY; BZ; CA; CH; CN; CO; CR; CU;
CZ; DE; DK; DM; DZ; EC; EE; EG; ES; FI;
GB; GD; GE; GH; GM; HR; HU; ID; IL; IN;
IS; JP; KE; KG; KP; KR; KZ; LC; LK; LR;
LS; LT; LU; LV; MA; MD; MG; MK; MN; MW;
MX; MZ; NA; NI; NO; NZ; OM; PG; PH; PL;
PT; RO; RU; SC; SD; SE; SG; SK; SL; SY;
TJ; TM; TN; TR; TT; TZ; UA; UG; US; UZ;
VC; VN; YU; ZA; ZM; ZW;

[EP] AT; BE; BG; CH; CY; CZ; DE; DK; EE; ES;
FI; FR; GB; GR; HU; IE; IT; LU; MC; NL;
PL; PT; RO; SE; SI; SK; TR;

[OA] BF; BJ; CF; CG; CI; CM; GA; GN; GQ; GW;
ML; MR; NE; SN; TD; TG;

[AP] BW; GH; GM; KE; LS; MW; MZ; NA; SD; SL;
SZ; TZ; UG; ZM; ZW;

[EA] AM; AZ; BY; KG; KZ; MD; RU; TJ; TM;

Publication Language: English
Filing Language: English
Fulltext word count: 15948

Detailed Description:

...generates source code in a high level language for each piece (process) of the generated **execution plan** 14. The code generator 16 also compiles the source code for execution in a target... ...of DBMS 1 0 include a query parser 13, a query
analyzer 15 and a **query optimizer** component 17. The **query** parser component 13 takes a native SQL query 12 and produces a parse tree structure... ...known techniques. The query analyzer component 15 takes the parse tree and produces an initial **query execution plan**. In turn, the **query optimizer** component 17 examines the initial **query execution plan** in conjunction with various runtime statistics and alters the **plan** to reduce its estimated compilation and execution costs. The result is an optimized high level **execution plan** 14 formed of a sequence of "pieces" which are targeted at different execution locales.

11/3K/16 (Item 14 from file: 349) Links
PCT FULLTEXT

01068600
OPTIMIZED DATABASE APPLIANCE
SYSTEME OPTIMISE DE BASE DE DONNEES

Patent Applicant/Patent Assignee:

- **NETEZZA CORPORATION**; 200 Crossing Boulevard, Framingham, MA 01702-4480
  US; US(Residence); US(Nationality)
  (For all designated states except: US)
- **HINSHAW Foster D**; 22 Campbell Park, Somerville, MA 02144
  US; US(Residence); US(Nationality)
  (Designated only for: US)
- **METZGER John K**; 2 Hundreds Road, Westborough, MA 01581
  US; US(Residence); US(Nationality)
  (Designated only for: US)
- **ZANE Barry M**; 4 Cobblestone Circle, Wayland, MA 01778
  US; US(Residence); US(Nationality)
  (Designated only for: US)

Patent Applicant/Inventor:

- **HINSHAW Foster D**
  22 Campbell Park, Somerville, MA 02144; US; US(Residence); US(Nationality); (Designated only for: US)
- **METZGER John K**
  2 Hundreds Road, Westborough, MA 01581; US; US(Residence); US(Nationality); (Designated only for: US)
- **ZANE Barry M**
  4 Cobblestone Circle, Wayland, MA 01778; US; US(Residence); US(Nationality); (Designated only for: US)

Legal Representative:

- **THIBODEAU David J Jr(et al)(agent)**
  Hamilton, Brook, Smith & Reynolds, P.C., 530 Virginia Road, P.O. Box 9133, Concord, MA 01742-9133; US;

	Country	Number	Kind	Date
Patent	WO	200398478	A1	20031127
Application	WO	2003US14790		20030509
Priorities	US	2002145571		20020513

Designated States: (All protection types applied unless otherwise stated - for applications 2004+)

[EP] AT; BE; BG; CH; CY; CZ; DE; DK; EE; ES;
FI; FR; GB; GR; HU; IE; IT; LU; MC; NL;
PT; RO; SE; SI; SK; TR;

[OA] BF; BJ; CF; CG; CI; CM; GA; GN; GQ; GW;
ML; MR; NE; SN; TD; TG;

[AP] GH; GM; KE; LS; MW; MZ; SD; SL; SZ; TZ;
UG; ZM; ZW;

[EA] AM; AZ; BY; KG; KZ; MD; RU; TJ; TM;

Publication Language: English
Filing Language: English
Fulltext word count: 6183

Detailed Description:

...other configuration.

The present invention provides for many dynamic operations. In one particular embodiment, a **query optimizer** tracks performance results from queries and reallocates the database tables and records in order to create improved query 5 execution. The execution engines dynamically track performance results from queries and **modify** the **query execution plan** to improve performance. The execution engines can also be dynamically reconfigured to process different database... ...database query 250. If the syntax is correct a query tree is produced, otherwise an **error** is generated.

A query optimizer 254 takes the query tree as input and generates an... ...possible steps, possible locales and data table access variants for each step of the execution **plan**.

Primary database catalog 256 stores the performance characteristics of the various components and the locality... ...table definitions, storage methods and database record count statistics. Using these characteristics and locality infonnation, **query optimizer** 254 can analyze the costs associated with various **execution plans** in order to choose the optimal **plan**. The optimal **execution plan** is generally considered to be the **execution plan** that will take the least elapsed time to complete. Other definitions of optimal (e.g., minimize 1/0 seeks or fabric traffic) are possible and the **query optimizer** 254 can be configured to create 1 5 **execution plans** based on alternative definitions of optimal.

Execution plan 258 is comprised of various parts (snippets). Snippets represent one or more database operations (e...

Claims:

...engine or the second execution engine dynamically track performance results from the executing queries and **modify** the query execution plan to improveperformance.
22 The system of Claim 1 wherein the... ...a database query;
parsing and validating the database query to produce a query tree; and**optimizing** the **query** tree to produce a locality-based execution planby accessing a database catalog comprising database table localityinformation, record locality information and execution engine information,the locality-based **execution plan** containing instructions for a centraldatabase operation processor providing a first execution engine to execute the locality-based **execution plan** by perforining at least a portion of the locality-based database operations and distributing at... ...engine or the second execution engine.
26 An apparatus for generating a distributed locality-based **execution plan** comprising:means for receiving a database query;means for parsing and validating the database query to produce aquery tree; andmeans for **optimizing** the **query** tree to produce a locality-based**execution plan** by accessing a database catalog comprising database tablelocality information, record locality information and execution engineinformation, the locality-based **execution plan** containing instructions for a 1 0 central database operation processor providing a first execution engine to execute the locality-based **execution plan** by performing at least a portion of the locality-based database operations and distributing at...

11/3K/18 (Item 16 from file: 349) Links
PCT FULLTEXT

00772911

A DATABASE SYSTEM FOR VIEWING EFFECTS OF CHANGES TO A INDEX FOR A QUERY OPTIMIZATION PLAN

SYSTEME DE BASE DE DONNEES POUR VISUALISER L'EFFET DE CHANGEMENTS D'UN INDEX POUR UN PLAN D'OPTIMISATION D'INTERROGATION

Patent Applicant/Patent Assignee:

- **COMPUTER ASSOCIATES THINK INC**; One Computer Associates Plaza, Islandia, NY 11749 US; US(Residence); US(Nationality)

Legal Representative:

- **FLIESLER Martin C**
  Fliesler Dubb Meyer and Lovejoy LLP, Suite 400, Four Embarcadero Center, San Francisco, CA 94111-4156; US;

	Country	Number	Kind	Date
Patent	WO	200106417	A1	20010125
Application	WO	2000US40424		20000719
Priorities	US	99356797		19990720

Designated States: (All protection types applied unless otherwise stated - for applications 2004+)

[EP] AT; BE; CH; CY; DE; DK; ES; FI; FR; GB; GR; IE; IT; LU; MC; NL; PT; SE;

[OA] BF; BJ; CF; CG; CI; CM; GA; GN; GW; ML; MR; NE; SN; TD; TG;

[AP] GH; GM; KE; LS; MW; MZ; SD; SL; SZ; TZ; UG; ZW;

[EA] AM; AZ; BY; KG; KZ; MD; RU; TJ; TM;

Publication Language: English
Filing Language: English
Fulltext word count: 7479

Detailed Description:

...replaced with references to the virtual table. The database management system then determines a new **optimization plan** for the **query**. Because the new **optimization** plan is determined using the virtual table and virtual index, the plan is retrieved much ... ...was excluded when the original table was copied to define the virtual table. Thus, any **changes** to the optimization **plan** may be identified quickly after the indexing design is altered.

Before the new optimization plan...

[File 347] **JAPIO** Dec 1976-2006/Dec(Updated 070403)

[File 350] **Derwent WPIX** 1963-2007/UD=200730

**File 350: DWPI has been enhanced to extend content and functionality of the database. For more info, visit http://www.dialog.com/dwpi/.*

; d s

Set	Items	Postings	Description
S1	475	4570	S (QUERY OR QUERIES)(3N)(OPTIMIZ??? OR OPTIMIS??? OR OPTIMIZATION OR OPTIMISATION)
S2	1026	8093	S (QUERY OR EXECUTION)(3N)PLAN? ? OR QAP OR QEP
S3	736372	2621081	S ERROR? ? OR FAIL???? OR FAULT? ? OR FUNCTION()CHECK???
S4	24468	77833	S (S2 OR PLAN? ? OR EXECUT???)(3N)(HALT??? OR STOP???? OR CEAS??? OR ABORT??? OR END???)
S5	10989	44790	S (PLAN? ? OR QAP OR QEP)(3N)(REBUILT OR REBUILD??? OR RE()(BUILT OR BUILD???) OR CORRECT??? OR MODIFY??? OR MODIFIE? ? OR MODIFICATION OR CHANG??? OR AMEND??? OR ALTER??? OR ALTERATION? ? OR EDIT??? OR REOPTIMIS? OR REOPTIMIZ?)
S6	4	24	S (MIDQUERY OR MID()QUERY OR SELFHEAL??? OR SELF()HEAL???)(3N)(REOPTIMIS? OR REOPTIMIZ? OR OPTIMIZ??? OR OPTIMIS??? OR OPTIMIZATION OR OPTIMISATION)
S7	3	231	S S1 AND S2 AND S4
S8	7	255	S S7 OR S6
S9	15	887	S S1 AND S2 AND S5
S10	15	887	S S9 NOT S8
S11	12	734	S S10 NOT AD=20040108:20070518/PR
S12	6	412	S S1 AND S2 AND S3
S13	3	151	S S1:S2 AND S3 AND S4
S14	8	561	S S12:S13
S15	3	64	S S14 NOT (S8 OR S10)

8/5/1 (Item 1 from file: 350) **Links**
Derwent WPIX

0015182003 *Drawing available*
WPI Acc no: 2005-531595/200554
XRPX Acc No: N2005-435191
Automatic error handling method in database engine of database management system, involves automatically rebuilding query access plan to generate new query access plan is response to error detected while executing access plan
Patent Assignee: IBM UK LTD (IBMC); INT BUSINESS MACHINES CORP (IBMC)
Inventor: DAY P R; MURAS B R; RYG A M

Patent Family (2 patents, 106 countries)

Patent Number	Kind	Date	Application Number	Kind	Date	Update	Type
US 20050154740	A1	20050714	US 2004754010	A	20040108	200554	B
WO 2005069163	A1	20050728	WO 2004EP53311	A	20041207	200554	E

Priority Applications (no., kind, date): US 2004754010 A 20040108

Patent Details

Patent Number	Kind	Lan	Pgs	Draw	Filing Notes	
US 20050154740	A1	EN	9	3		
WO 2005069163	A1	EN				
National Designated States,Original	AE AG AL AM AT AU AZ BA BB BG BR BW BY BZ CA CH CN CO CR CU CZ DE DK DM DZ EC EE EG ES FI GB GD GE GH GM HR HU ID IL IN IS JP KE KG KP KR KZ LC LK LR LS LT LU LV MA MD MG MK MN MW MX MZ NA NI NO NZ OM PG PH PL PT RO RU SC SD SE SG SK SL SY TJ TM TN TR TT TZ UA UG US UZ VC VN YU ZA ZM ZW					
Regional Designated States,Original	AT BE BG BW CH CY CZ DE DK EA EE ES FI FR GB GH GM GR HU IE IS IT KE LS LT LU MC MW MZ NA NL OA PL PT RO SD SE SI SK SL SZ TR TZ UG ZM ZW					

Alerting Abstract US A1
NOVELTY - A query access plan is automatically rebuilt to generate a new query access plan in response to error detected while executing the access plan and the new query access plan is executed.
DESCRIPTION - INDEPENDENT CLAIMS are also included for the following:

- program product for automatic handling of errors; and
- computer.

USE - For automatically handling errors in database engine of database management system (DBMS) in networked computer (claimed) e.g. client computer, server computer, handheld computer, embedded controller connected through networks such as local area network (LAN), wide area network (WAN), internet.
ADVANTAGE - Automatically addresses the errors without the need for customer support thereby avoiding performance degradations.
DESCRIPTION OF DRAWINGS - The figure shows a block diagram of the network computer system including DBMS.
10,20 computers

8/5/3 (Item 3 from file: 350) **Links**
Derwent WPIX

0014672546 *Drawing available*
WPI Acc no: 2005-020127/200502
XRPX Acc No: N2005-017104
Live switchover implementing method for use in database management system, involves identifying temporary sparse index, stopping processing of query with plan, and providing live switchover to temporary sparse index
Patent Assignee: INT BUSINESS MACHINES CORP (IBMC)
Inventor: DAY P R; MURAS B R

Patent Family (2 patents, 1 countries)

Patent Number	Kind	Date	Application Number	Kind	Date	Update	Type
US 20040236727	A1	20041125	US 2003443921	A	20030522	200502	B
US 7191174	B2	20070313	US 2003443921	A	20030522	200721	E

Priority Applications (no., kind, date): US 2003443921 A 20030522

Patent Details

Patent Number	Kind	Lan	Pgs	Draw	Filing Notes
US 20040236727	A1	EN	12	6	

Alerting Abstract US A1
NOVELTY - The method involves using a plan with an existing index for processing a query. A temporary sparse index is built simultaneously with the plan. The sparse index is identified, processing of the **query** with the **plan** is **stopped**, and a live switchover is provided to the sparse index. Another plan with the sparse index is used to continue the query processing at a point where the query processing is **stopped** with the former **plan**.
DESCRIPTION - INDEPENDENT CLAIMS are also included for the following:

- a **query optimizer** for implementing **query** performance
- a computer program product for implementing enhanced query performance in a computer system.

USE - Used for implementing live switchover to a temporary sparse index in a database management system.
ADVANTAGE - The method implements live switchover to a temporary sparse index for faster query performance, thus improving and **optimizing queries** in a database management system.
DESCRIPTION OF DRAWINGS - The drawing shows a flowchart illustrating the steps performed by a computer system for implementing a method to process and **optimize** database **query**.

11/5/1 (Item 1 from file: 350) **Links**
Derwent WPIX

0014997314 *Drawing available*
WPI Acc no: 2005-345198/200535
XRPX Acc No: N2005-282105
Computer-implemented method for accelerating database query processing, involves reoptimizing query and restarting its execution with reoptimized query plan, if continuous execution of particular query execution plan is not worthwhile
Patent Assignee: LOHMAN G M (LOHM-I); PIRAHESH M H (PIRA-I); RAMAN V (RAMA-I); SIMMEN D E (SIMM-I); VOLKER M (VOLK-I)
Inventor: LOHMAN G M; PIRAHESH M H; RAMAN V; SIMMEN D E; VOLKER M

Patent Family (1 patents, 1 countries)

Patent Number	Kind	Date	Application Number	Kind	Date	Update	Type
US 20050097078	A1	20050505	US 2003698828	A	20031031	200535	B

Priority Applications (no., kind, date): US 2003698828 A 20031031

Patent Details

Patent Number	Kind	Lan	Pgs	Draw	Filing Notes
US 20050097078	A1	EN	29	17	

Alerting Abstract US A1
NOVELTY - The method involves determining that continuous execution of particular **query execution plan** is not worthwhile, when significant amount of query execution remains, and significant parameter estimation errors occur when computing difference of estimated and actual **optimization** parameter values. The **query** is reoptimized and query execution is restarted with **reoptimized query plan**, if continuous **execution** is not worthwhile.
DESCRIPTION - INDEPENDENT CLAIMS are also included for the following:

- computer implemented system for accelerating database query processing; and
- computer program product tangibly embodying program for accelerating database query processing.

USE - For accelerating database query processing in progressive query processing architecture.
ADVANTAGE - Enables performing query processing in a robust manner in the face of optimization errors.
DESCRIPTION OF DRAWINGS - The figure shows a block diagram of the progressive query processing architecture.

11/5/5 (Item 5 from file: 350) **Links**
Derwent WPIX

0010918747 *Drawing available*
WPI Acc no: 2001-540391/200160
XRPX Acc No: N2001-401523
Query-plan modification method for computer implemented database management system, involves identifying generated transient views improving performance of query-plan
Patent Assignee: INT BUSINESS MACHINES CORP (IBMC)
Inventor: SUBRAMANIAN N I; VENKATARAMAN S

Patent Family (1 patents, 1 countries)

Patent Number	Kind	Date	Application Number	Kind	Date	Update	Type
US 6275818	B1	20010814	US 199763979	P	19971106	200160	B
			US 1998186804	A	19981105		

Priority Applications (no., kind, date): US 199763979 P 19971106; US 1998186804 A 19981105

Patent Details

Patent Number	Kind	Lan	Pgs	Draw	Filing Notes	
US 6275818	B1	EN	24	11	Related to Provisional	US 199763979

Alerting Abstract US B1
NOVELTY - Equivalence classes containing similar sub-**plans** of **query-plan** are generated. Transient views containing a union of results from all equivalence class associated sub-plans, are generated. The transient views improving **query-plan** performance are identified and results for each associated equivalence class sub-plan, are obtained by filtering the identified views.
DESCRIPTION - An INDEPENDENT CLAIM is also included for **queries optimizing** apparatus.
USE - For **optimizing queries** in computer implemented database management system.
ADVANTAGE - By identifying transient views improving **query plan**, new **query plan** resulting in improved **query** performance is efficiently generated. Provides efficient **query optimizing** technique that can be implemented over existing query processing system in non-intrusive manner.
DESCRIPTION OF DRAWINGS - The figure shows the flow diagram illustrating the steps performed by cost based optimizer to identify execution steps for efficient query processing.

11/5/6 (Item 6 from file: 350) **Links**
Derwent WPIX

0010760503 *Drawing available*
WPI Acc no: 2001-374128/200139
XRPX Acc No: N2001-273750
Changes viewing method for index of query optimization plan in database management systems, involves determining new optimization plan for query by adding reference to virtual index to query
Patent Assignee: COMPUTER ASSOC THINK INC (COMP-N); PLATINUM TECHNOLOGY IP INC (PLAT-N)
Inventor: KOSCIUSZKO E; MENON S; VO H; VO H V

Patent Family (15 patents, 91 countries)

Patent Number	Kind	Date	Application Number	Kind	Date	Update	Type
WO 2001006417	A1	20010125	WO 2000US40424	A	20000719	200139	B
AU 200071357	A	20010205	AU 200071357	A	20000719	200139	E
US 6317736	B1	20011113	US 1999356797	A	19990720	200173	E
US 20020010701	A1	20020124	US 1999356797	A	19990720	200210	E
			US 2001900791	A	20010706		
BR 200012684	A	20020416	BR 200012684	A	20000719	200234	E
			WO 2000US40424	A	20000719		
EP 1206746	A1	20020522	EP 2000960157	A	20000719	200241	E
			WO 2000US40424	A	20000719		
KR 2002031390	A	20020501	KR 2002700823	A	20020119	200270	E
CN 1361890	A	20020731	CN 2000810565	A	20000719	200279	E
JP 2003505766	W	20030212	WO 2000US40424	A	20000719	200321	E
			JP 2001511602	A	20000719		
ZA 200200389	A	20030326	ZA 2002389	A	20020116	200327	E
US 6560593	B1	20030506	US 1999356797	A	19990720	200338	E
AU 768487	B	20031211	AU 200071357	A	20000719	200404	E
US 6898588	B2	20050524	US 1999356797	A	19990720	200535	E
			US 2001900791	A	20010706		
IN 200200034	P3	20060505	WO 2000US40424	A	20000719	200643	E
			IN 2002MN34	A	20020110		
IL 147694	A	20070211	IL 147694	A	20000719	200719	E

Priority Applications (no., kind, date): US 1999356797 A 19990720; US 2001900791 A 20010706

Patent Details

Patent Number	Kind	Lan	Pgs	Draw	Filing Notes	
WO 2001006417	A1	EN	33	4		
National Designated States,Original	AE AG AL AM AT AU AZ BA BB BG BR BY CA CH CN CR CU CZ DE DK DM DZ EE ES FI GB GD GE GH GM HU ID IL IN IS JP KE KG KP KR KZ LC LK LR LS LT LU LV MA MD MG MK MN MW MX NO NZ PL PT RO RU SD SE SG SI SK SL TJ TM TR TT TZ UA UG UZ VN YU ZA ZW					
Regional Designated States,Original	AT BE CH CY DE DK EA ES FI FR GB GH GM GR IE IT KE LS LU MC MW MZ NL OA PT SD SE SL SZ TZ UG ZW					
AU 200071357	A	EN			Based on OPI patent	WO 2001006417
US 20020010701	A1	EN			Continuation of application	US 1999356797
BR 200012684	A	PT			PCT Application	WO 2000US40424
					Based on OPI patent	WO 2001006417
EP 1206746	A1	EN			PCT Application	WO 2000US40424
					Based on OPI patent	WO 2001006417

Regional Designated States,Original	AL AT BE CH CY DE DK ES FI FR GB GR IE IT LI LT LU LV MC MK NL PT RO SE SI					
JP 2003505766	W	JA	40		PCT Application	WO 2000US40424
					Based on OPI patent	WO 2001006417
ZA 200200389	A	EN	42			
AU 768487	B	EN			Previously issued patent	AU 200071357
					Based on OPI patent	WO 2001006417
US 6898588	B2	EN			Continuation of application	US 1999356797
					Continuation of patent	US 6560593
IN 200200034	P3	EN			PCT Application	WO 2000US40424
IL 147694	A	EN			Based on OPI patent	WO 2001006417

Alerting Abstract WO A1

NOVELTY - An original table excluding data to define a virtual table is copied and a virtual index associated with virtual table is provided. Reference to original table in a query is replaced with reference to virtual table and the reference to the virtual index is added to the query to determine a new **optimization plan** for the **query**.

DESCRIPTION - The changes to an original optimization plan for quency having reference to original table with data stored in database is viewed.

USE - In database management system for determining **optimization** plans for database **queries**.

ADVANTAGE - Virtual index is easily and quickly modified while preserving the overall structure of the original table due to exclusion of data when copying the original table to define the virtual table. User need not know about the use of virtual objects in creating new optimization plan because of the replacement of references. New indices may be added and existing indices may be dropped very quickly and also new virtual index is easily created even when no original index exists. Since new optimization plan is determined using virtual table and virtual index, the plan is referenced much faster than if it is were created using original table and any associated original index, thus any **changes** to optimization **plan** is identified quickly after the indexing design is altered.

DESCRIPTION OF DRAWINGS - The figure shows a flow diagram of the method for viewing changes to an index of a database table.

11/5/7 (Item 7 from file: 350) **Links**
Derwent WPIX

0009806805 *Drawing available*
WPI Acc no: 2000-096456/200008
Related WPI Acc No: 1996-518229
XRPX Acc No: N2000-074481
Computer database data retrieval program to process a query statement to identify data to be retrieved
Patent Assignee: INT BUSINESS MACHINES CORP (IBMC)
Inventor: HUANG D T; LIN E T; WANG Y

Patent Family (1 patents, 1 countries)

Patent Number	Kind	Date	Application Number	Kind	Date	Update	Type
US 6009265	A	19991228	US 1994201822	A	19940225	200008	B
			US 1995486087	A	19950607		
			US 1996617003	A	19960314		

Priority Applications (no., kind, date): US 1995486087 A 19950607; US 1994201822 A 19940225; US 1996617003 A 19960314

Patent Details

Patent Number	Kind	Lan	Pgs	Draw	Filing Notes	
US 6009265	A	EN	14	7	Division of application	US 1994201822
					Continuation of application	US 1995486087

Alerting Abstract US A
NOVELTY - During compile time the program:

- determines an optimal sequential execution plan for the **query statement**, and
- determines **an** optimal parallel execution plan based on the optimal **sequential execution** plan, and based on available **resource information**;

and during run time:

1. determines, based on a value of at least one run time variable, whether to use the sequential execution plan without modification or a modification of the parallel execution plan **as an** optimal **run** time execution plan, and
2. executes the **query statement** using the optimal run time **execution plan.**
3. USE - As a database management system (DBMS).
   ADVANTAGE - Optimizes the execution of a database query by determining a parallel execution strategy during bind time to minimize consumption of execution time while allowing some flexibility to make **changes** in the **plan** in response to the run time environment.
   DESCRIPTION OF DRAWINGS - The drawing shows a high level flowchart of the **query optimization**.

[File 2] **INSPEC** 1898-2007/May W1

[File 6] **NTIS** 1964-2007/May W3

[File 8] **Ei Compendex(R)** 1884-2007/May W1

[File 23] **CSA Technology Research Database** 1963-2007/May

[File 34] **SciSearch(R) Cited Ref Sci** 1990-2007/May W3

[File 35] **Dissertation Abs Online** 1861-2007/Apr

[File 65] **Inside Conferences** 1993-2007/May 18

[File 95] **TEME-Technology & Management** 1989-2007/May W2

[File 99] **Wilson Appl. Sci & Tech Abs** 1983-2007/Apr

[File 111] **TGG Natl.Newspaper Index(SM)** 1979-2007/May 15

[File 144] **Pascal** 1973-2007/Apr W5

[File 239] **Mathsci** 1940-2007/Jun

[File 256] **TecInfoSource** 82-2007/Jun

[File 434] **SciSearch(R) Cited Ref Sci** 1974-1989/Dec

;d s

Set	Items	Postings	Description
S1	8549	34351	S (QUERY OR QUERIES)(3N)(OPTIMIZ??? OR OPTIMIS??? OR OPTIMIZATION OR OPTIMISATION)
S2	6385	18367	S (QUERY OR EXECUTION)(3N)PLAN? ? OR QAP OR QEP
S3	3928454	8705069	S ERROR? ? OR FAIL???? OR FAULT? ? OR FUNCTION()CHECK???
S4	12110	26195	S (S2 OR PLAN? ? OR EXECUT???)(3N)(HALT??? OR STOP???? OR CEAS??? OR ABORT??? OR END???)
S5	50075	113801	S (PLAN? ? OR QAP OR QEP)(3N)(REBUILT OR REBUILD??? OR RE()(BUILT OR BUILD???) OR CORRECT??? OR MODIFY??? OR MODIFIE? ? OR MODIFICATION OR CHANG??? OR AMEND??? OR ALTER??? OR ALTERATION? ? OR EDIT??? OR REOPTIMIS? OR REOPTIMIZ?)
S6	174	558	S (MIDQUERY OR MID()QUERY OR SELFHEAL??? OR SELF()HEAL???)(3N)(REOPTIMIS? OR REOPTIMIZ? OR OPTIMIZ??? OR OPTIMIS??? OR OPTIMIZATION OR OPTIMISATION)
S7	1	18	S S1 AND S2 AND S3 AND S4
S8	12	228	S S1 AND S2 AND S3:S4 AND S5

S9	12	228	S S8 NOT S7
S10	5	97	RD (unique items)
S11	12	279	S S1:S2 AND S6
S12	3	57	RD (unique items)
S13	2	34	S S12 NOT (S7 OR S10)

10/5/3 (Item 3 from file: 2) **Links**
INSPEC

06484858 **INSPEC Abstract Number:** B9703-6210L-023, C9703-6160B-013
Title: Scrambling query plans to cope with unexpected delays
Author Amsaleg, L.; Tomasic, A.; Franklin, M.J.; Urhan, T.
Author Affiliation: Maryland Univ., MD, USA
Conference Title: Proceedings of the Fourth International Conference on Parallel and Distributed Information Systems (Cat. No.96TB100085) p. 208-19
Publisher: IEEE Comput. Soc. Press , Los Alamitos, CA, USA
Publication Date: 1996 **Country of Publication:** USA xi+295 pp.
ISBN: 0 8186 7475 X **Material Identity Number:** XX96-03187
U.S. Copyright Clearance Center Code: 0 8186 7475 X/96/$5.00
Conference Title: Proceedings of 4th International Conference on Parallel and Distributed Information Systems
Conference Sponsor: IEEE Compt. Soc. Tech. Committee on Data Eng.; ACM SIGMOD
Conference Date: 18-20 Dec. 1996 **Conference Location:** Miami Beach, FL, USA

Language: English **Document Type:** Conference Paper (PA)
Treatment: Practical (P)
Abstract: Accessing data from numerous widely distributed sources poses significant new challenges for **query optimization** and execution. Congestion and **failures** in the network can introduce highly variable response times for wide area data access. The paper is an initial exploration of solutions to this variability. We introduce a class of dynamic, run time **query plan modification** techniques that we call **query plan** scrambling. We present an algorithm that **modifies execution plans** on-the-fly in response to unexpected delays in obtaining initial requested tuples from remote sources. The algorithm both reschedules operators and introduces new operators into the **query plan**. We present simulation results that demonstrate how the technique effectively hides delays by performing other useful work while waiting for missing data to arrive. (18 Refs)
Subfile: B C
Descriptors: distributed databases; query processing; wide area networks
Identifiers: unexpected delays; data access; widely distributed sources; **query optimization**; highly variable response times; wide area data access; run time **query plan modification** techniques; **query plan** scrambling; initial requested tuples; remote sources; missing data
Class Codes: B6210L (Computer communications); C6160B (Distributed databases); C4250 (Database theory); C5620W (Other computer networks)

10/5/4 (Item 1 from file: 8) Links
Fulltext available through: USPTO Full Text Retrieval Options
Ei Compendex(R)

08085925 **E.I. No:** EIP98084331474
Title: Dynamic query operator scheduling for wide-area remote access

Author: Amsaleg, Laurent; Franklin, Michael J.; Tomasic, Anthony
Corporate Source: IRISA/INRIA, Rennes, Fr
Source: Distributed and Parallel Databases v 6 n 3 Jul 1998. p 217-246
Publication Year: 1998
CODEN: DAATES **ISSN:** 0926-8782
Language: English
Document Type: JA; (Journal Article) **Treatment:** A; (Applications); T; (Theoretical)
Journal Announcement: 9810W1
Abstract: Distributed databases operating over wide-area networks such as the Internet, must deal with the unpredictable nature of the performance of communication. The response times of accessing remote sources can vary widely due to network congestion, link **failure**, and other problems. In such an unpredictable environment, the traditional iterator-based query execution model performs poorly. We have developed a class of methods, called query scrambling, for dealing explicitly with the problem of unpredictable response times. Query scrambling dynamically **modifies query execution plans** on-the-fly in reaction to unexpected delays in data access. In this paper we focus on the dynamic scheduling of query operators in the context of query scrambling. We explore various choices for dynamic scheduling and examine, through a detailed simulation, the effects of these choices. Our experimental environment considers pipelined and non-pipelined join processing in a client with multiple remote data sources and delayed or possibly bursty arrivals of data. Our performance results show that scrambling rescheduling is effective in hiding the impact of delays on query response time for a number of different delay scenarios. (Author abstract) 23 Refs.
Descriptors: *Database systems; Distributed computer systems; Wide area networks; Pipeline processing systems; Scheduling; Mathematical models; Computer simulation; Optimization; Performance
Identifiers: Dynamic query operator scheduling; Wide area remote access; Distributed **query** processing; Dynamic **query optimization**; Internet
Classification Codes:
723.3 (Database Systems); 722.4 (Digital Computers & Systems); 722.3 (Data Communication, Equipment & Techniques); 921.6 (Numerical Methods); 723.5 (Computer Applications); 921.5 (Optimization Techniques)
723 (Computer Software); 722 (Computer Hardware); 921 (Applied Mathematics)
72 (COMPUTERS & DATA PROCESSING); 92 (ENGINEERING MATHEMATICS)

02069976 ORDER NO: AADAA-I3165339
Handling estimation errors in database query processing

Author: Deshpande, Amol Vishnupant
Degree: Ph.D.
Year: 2004
Corporate Source/Institution: University of California, Berkeley (0028)
Chair: Joseph M. Hellerstein
Source: Volume 6602B of Dissertations Abstracts International.
PAGE 981 . 298 PAGES
Descriptors: COMPUTER SCIENCE
Descriptor Codes: 0984
ISBN: 0-542-00764-9

A **query optimizer** is among the core pieces of a modern database management system, responsible for choosing a **query execution plan** for a user-provided declarative **query**. **Query optimizers** typically employ a cost estimation procedure to compare costs of different **execution plans**. **Errors** in this estimation process are quite common and arise due to reasons such as incomplete and insufficient statistical information about the data, and highly variable runtime environments that can affect the plan costs in unpredictable manners. Two approaches have been previously proposed for handling such estimation **errors**: (1) building sophisticated synopsis techniques that succintly summarize the data in the database and thus provide more statistical information to the **query optimizer**, and (2) aggressive reoptimization schemes that attempt to **change** the **execution plans** chosen to execute queries, on-the-fly.

In the first part of this dissertation, we focus on building and using sophisticated synopsis techniques in the context of a traditional **query optimizer**. We propose a class of synopsis techniques called D<smcap>EPENDENCY</smcap>-B<smcap> ASED</smcap> Histograms that use statistical interaction models to exploit the correlations in the data, and to estimate selectivities efficiently. We also develop an efficient algorithm to search through the class of statistical models that we employ. Using sophisticated synopsis techniques such as these in the context of a traditional **query optimizer** poses interesting computational challenges; a naive approach to doing this could make the **query optimization** process so expensive as to be ineffective. This naturally leads to an “estimation planning” problem that asks for the best strategy to compute all the estimates required by an optimizer using the synopses at its disposal. We analyze this problem, its solution space, and propose algorithms to efficiently find good estimation plans.

There are many scenarios where sophisticated synopsis techniques may not be applicable; examples include wide area and web based data sources, data streams and complex data domains. In the second part of this dissertation, we explore a highly-adaptive query processing technique called <italic>eddies </italic> that treats query processing as <italic>routing</italic> of tuples through operators, and adapts to changing data and runtime characteristics by continuously changing the order in which tuples are routed. We analyze the eddies architecture and identify a fundamental flaw in the basic design of the architecture: the <italic>burden of history</italic> in routing. (Abstract shortened by UMI.)

13/5/2 (Item 2 from file: 2) **Links**
Fulltext available through: ACM - Association for Computing Machinery USPTO Full Text Retrieval Options
INSPEC

06940197 **INSPEC Abstract Number:** C9807-6160J-007
Title: Efficient mid-query re-optimization of sub-optimal query execution plans
Author Kabra, N.; DeWitt, D.J.
Author Affiliation: Dept. of Comput. Sci., Wisconsin Univ., Madison, WI, USA
Journal: SIGMOD Record **Conference Title:** SIGMOD Rec. (USA) vol.27, no.2 p. 106-17
Publisher: ACM ,
Publication Date: June 1998 **Country of Publication:** USA
CODEN: SRECD8 **ISSN:** 0163-5808
SICI: 0163-5808(199806)27:2L.106:EQOO;1-V
Material Identity Number: A660-98003
Conference Title: 1998 ACM SIGMOD International Conference on Management of Data
Conference Date: 1-4 June 1998 **Conference Location:** Seattle, WA, USA

Language: English **Document Type:** Conference Paper (PA); Journal Paper (JP)
Treatment: Practical (P)
Abstract: For a number of reasons, even the best **query optimizers** can very often produce sub optimal **query execution plans**, leading to a significant degradation of performance. This is especially true in databases used for complex decision support queries and/or object relational databases. We describe an algorithm that detects sub optimality of a **query execution plan** during **query execution** and attempts to correct the problem. The basic idea is to collect statistics at key points during the execution of a complex query. These statistics are then used to optimize the execution of the query, either by improving the resource allocation for that query, or by changing the **execution plan** for the remainder of the query. To ensure that this does not significantly slow down the normal execution of a **query**, the **Query Optimizer** carefully chooses what statistics to collect, when to collect them, and the circumstances under which to re **optimize** the **query**. We describe an implementation of this algorithm in the Paradise Database System, and we report on performance studies, which indicate that this can result in significant improvements in the performance of complex queries. (27 Refs)
Subfile: C
Descriptors: decision support systems; object-oriented databases; query processing; relational databases; resource allocation
Identifiers: mid query re-**optimization**; sub optimal **query execution plans**; **query optimizers**; performance degradation; complex decision support queries; object relational databases; query execution; resource allocation; Paradise Database System; performance studies; complex queries
Class Codes: C6160J (Object-oriented databases); C4250 (Database theory); C6160D (Relational databases); C7102 (Decision support systems); C6150J (Operating systems)